



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/008,691	11/08/2001	Jay P. Hoeflinger	042390.P11920	7462
8791	7590	02/10/2005	EXAMINER	
BLAKELY SOKOLOFF TAYLOR & ZAFMAN 12400 WILSHIRE BOULEVARD SEVENTH FLOOR LOS ANGELES, CA 90025-1030			KISS, ERIC B	
			ART UNIT	PAPER NUMBER
			2122	

DATE MAILED: 02/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>		<b>Applicant(s)</b>	
	10/008,691		HOEFLINGER ET AL.	
	<b>Examiner</b>		<b>Art Unit</b>	
	Eric B. Kiss		2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 08 November 2001.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-26 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. Claims 1-26 have been examined.

#### *Specification*

2. The use of various trademarks has been noted in this application. Trademarks should be capitalized wherever they appear and be accompanied by generic terminology, *i.e.*, use trademarks as adjectives modifying an descriptive noun.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

The Examiner provides the following as mere suggestions for improving the specification, and Applicant is free to apply or ignore these suggestions at their discretion:

**“Intel® Corporation”** on page 4, paragraph [0015], line 6, should read **--Intel Corporation--** as in this instance, “Intel” is used as the name of the company and not descriptive of one of its products (*i.e.*, not a trademark).

On page 9, at the end of paragraph [0030], **“UNIX, Microsoft Windows™, and real time operating systems such as VxWorks™, etc.”** would be more appropriately worded as **--UNIX® operating systems, Microsoft® Windows® operating systems, and real time operating systems such as the VxWorks® operating system, etc.--**.

On page 8, in paragraph [0027], “**OpenMP**” would be more appropriately worded as --**OpenMP® specification**--.

On page 9, in paragraph [0031], “**OpenMP**” would be more appropriately worded as --**OpenMP® specification**--.

On page 14, in paragraph [0043], “**a set of directives other than OpenMP**” would be more appropriately worded as --**a set of directives other than those in an OpenMP® specification**--.

#### ***Claim Objections***

3. Claim 21 is objected to because of the following informalities: the period (.) at the end of link 14 should be replaced with --; and--. Appropriate correction is required.

#### ***Claim Rejections - 35 USC § 112***

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claim 23 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 23 recites the limitation "the entity" in line 1. There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, claim 23 is interpreted as being dependent from claim 22 (instead of claim 21) for the purpose of further examination.

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Working Draft: J3/00-007R3," October 2000 (hereinafter *Fortran2000*) in view of Alfred V. Aho, et al., "Compilers: Principles, Techniques, and Tools," 1988 (hereinafter *Aho et al.*).

As per claim 1, *Fortran2000* discloses generating a function having an argument, the function expressed in a high-level programming language, wherein the function includes a set of one or more instructions to return a memory address of the argument as a result of the function (see, for example, the description of *C\_LOC(X)* in section 16.2.3 on pages 395-396); and generating a call to the function, the call expressed in the high-level programming language,

Art Unit: 2122

wherein the call passes a descriptor as the argument (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396).

*Fortran2000* is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit to arrive at a machine-executable implementation of program source code (see, for example, “The Context of a Compiler” on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 2, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 3, *Fortran2000* further discloses the argument being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 4, *Fortran2000* discloses receiving a first code, wherein the first code refers to a variable of a target data type, wherein the variable is addressable using a descriptor (see, for example, section 16.2 on pages 392-401); and translating the first code into a second code, the second code expressed in a high-level programming language (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396), wherein the translation requires a memory address of the descriptor, and wherein the translation comprises: generating a function having an argument, wherein the function includes a set of one or more instructions that return the memory address of the argument as a result of the function (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396); and generating a call to the function, wherein the call passes the descriptor as the argument (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396).

*Fortran2000* is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit to arrive at a machine-executable implementation of program source code (see, for example, “The Context of a Compiler” on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 5, *Fortran2000* discloses the translating comprising generating an interface block for the function for each different target data type in the first code (see, for example, section 16.2 on pages 392-401). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 6, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 7, *Fortran2000* further discloses the argument being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 8, *Fortran2000* further discloses generating a data structure to store information based on the target data type (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396; the *C\_LOC (X)* function produces a *C\_PTR* scalar return value). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claims 9 and 10, *Fortran2000* further discloses the function including a routine from a runtime library written in a C programming language, the routine to return a memory



Art Unit: 2122

address of an argument of the routine (see, for example, section 16.2 on pages 392-401, and in particular, see the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such claims also would have been obvious.

As per claims 11-13, these are machine-readable medium versions of the claims methods discussed above (claims 1-3, respectfully). Further, the use of such a machine-readable medium is inherent in implementing the computer software methods disclosed in *Fortran2000*. Therefore, for reasons stated above, such claims also would have been obvious.

As per claims 14-20, these are machine-readable medium versions of the claims methods discussed above (claims 4-10, respectfully). Further, the use of such a machine-readable medium is inherent in implementing the computer software methods disclosed in *Fortran2000*. Therefore, for reasons stated above, such claims also would have been obvious.

As per claim 21, *Fortran2000* discloses a translation unit to receive a first code that refers to a variable of a target data type, wherein the variable is referred to by a descriptor (see, for example, section 16.2 on pages 392-401), the translation unit to translate the first code into a second code, the second code based on a high-level programming language (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396), wherein the translation requires a memory address of the descriptor and wherein the translation comprises: generating a function having an argument, wherein the function includes a set of one or more instructions to return the memory address of the argument as a result of the function (see, for example, the

Art Unit: 2122

description of *C\_LOC (X)* in section 16.2.3 on pages 395-396); and generating a call to the function, wherein the call passes the descriptor as the argument (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396);

*Fortran2000* is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit and a linker unit to arrive at a machine-executable implementation of program source code (see, for example, "The Context of a Compiler" on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit and a linker unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 22, *Fortran2000* further discloses the generation of the second code including the generation of a function, the function having an entity as an argument, and a call to the function, wherein the call to the function accepts the argument as an entity for which the memory address can be determined and returned as a result of the function (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396). The use of a compiler unit to provide the requisite functionality has been addressed as set forth above for claim 21. Therefore, for reasons stated above, such a claim also would have been obvious.

Art Unit: 2122

As per claim 23, *Fortran2000* further discloses the entity being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 24, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claims 25 and 26, *Fortran2000* further discloses the function including a routine from a runtime library written in a C programming language, the routine to return a memory address of an argument of the routine (see, for example, section 16.2 on pages 392-401, and in particular, see the description of *C\_LOC (X)* in section 16.2.3 on pages 395-396).

### ***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

9. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The

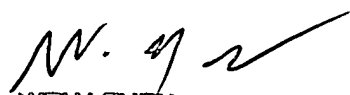
Art Unit: 2122

Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

EBK/EBK  
February 1, 2005

  
WEI Y. ZHEN  
PRIMARY EXAMINER